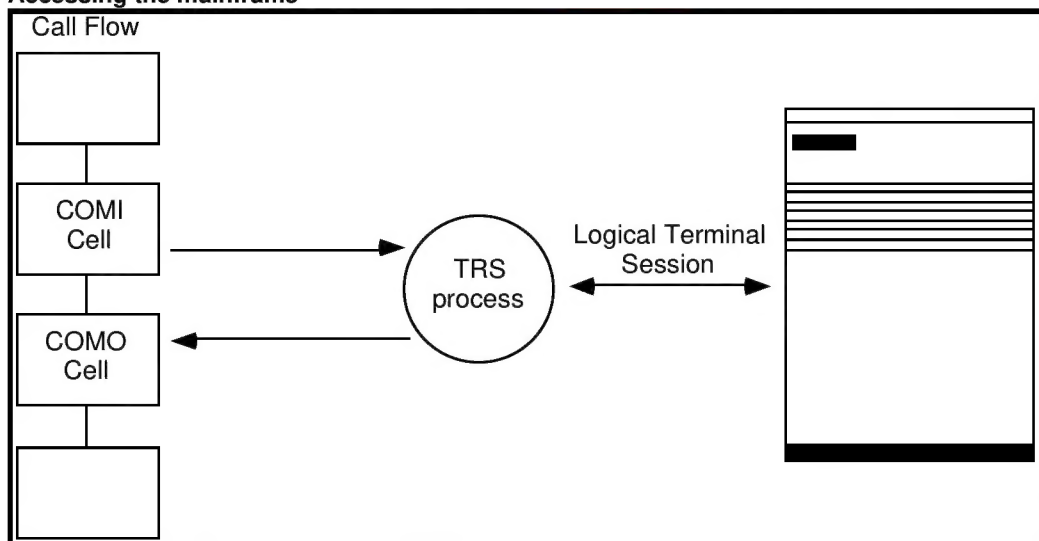


Chapter 4: Meridian IVR 2.0/I call flow interface

Now that you understand how to script host transactions as action and screen templates, you can begin integrating these templates into your Meridian IVR 2.0/I Applications (see [Figure 4-1](#)).

Figure 4-1
Accessing the mainframe



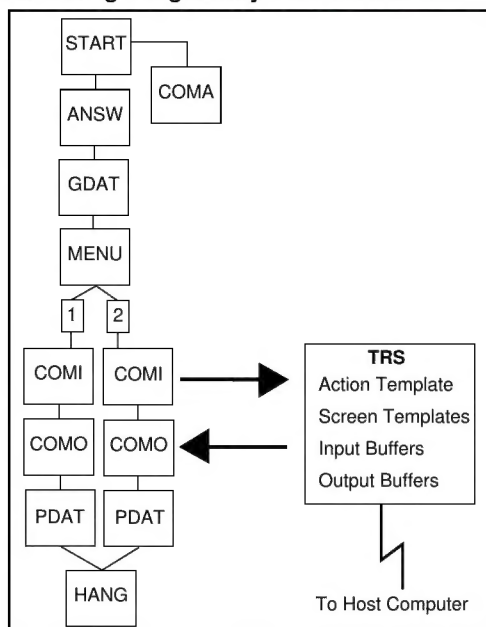
This chapter explains:

- How to code your applications to perform host transactions
- How to use the COMI, COMO, and COMA cells to communicate with the host computer

Using the COMI, COMO, and COMA cells to access the host computer

With Meridian IVR 2.0/I, the links between the Meridian IVR 2.0/I call flow and the TRS are the COM cells. The COMI cell sends input to the TRS process and identifies the action template file, the COMO cell receives output from the TRS process, and the COMA cell aborts a transaction in progress (usually used in the clean-up branch in case a caller hangs up before the transaction completes). **Figure 4-2** illustrates this.

Figure 4-2
Activating the gateway from a COMI cell



To develop an application that processes one or more terminal sessions, just start as if you were developing any other voice application:

- Determine the telephone interaction with the caller and create the corresponding Meridian IVR 2.0/I call flow.
- At the point in the call flow where you require interaction with the host computer, insert a COMI cell.
- Insert a COMO cell to receive the output from the TRS transaction.

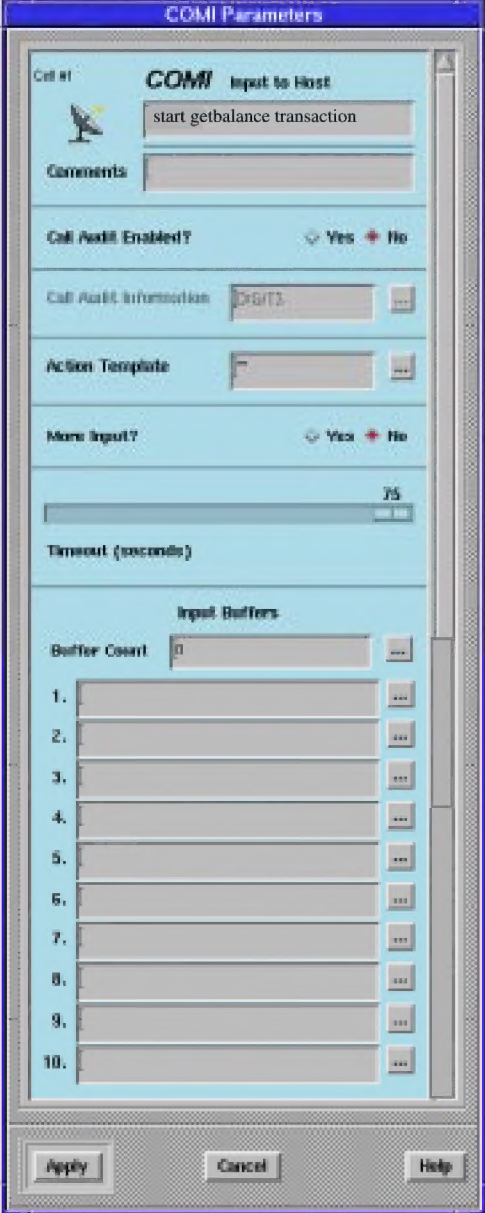
You must always follow a COMI cell with a COMO cell, even if you do not require the return of any data.

The following sections show you how to code each of these cells for use with the same example transaction presented in Chapter 3.

Setting the COMI cell parameters

Figure 4-3 shows the parameter window for the COMI cell.

Figure 4-3
COMI cell parameter window



The image shows a 'COMI Parameters' dialog box with a blue title bar. It contains several sections for configuring a COMI cell. At the top, 'Cell #1' is set to 'COMI' and 'Input to Host' is 'start getbalance transaction'. Below this is a 'Comments' field. A 'Call Audit Enabled?' section has radio buttons for 'Yes' and 'No', with 'No' selected. The 'Call Audit Information' field contains 'D+G/T3'. An 'Action Template' field is empty. Another 'More Input?' section has radio buttons for 'Yes' and 'No', with 'No' selected. A slider bar is set to '75'. The 'Timeout (seconds)' field is empty. The 'Input Buffers' section has a 'Buffer Count' of '0' and a list of 10 numbered input buffers, each with an empty text field and a button. At the bottom are 'Apply', 'Cancel', and 'Help' buttons.

COMI Parameters

Cell #1: **COMI** Input to Host: start getbalance transaction

Comments:

Call Audit Enabled? ☒ Yes ☐ No

Call Audit Information: D+G/T3

Action Template:

More Input? ☒ Yes ☐ No

75

Timeout (seconds):

Input Buffers

Buffer Count: 0

1.

2.

3.

4.

5.

6.

7.

8.

9.

10.

The following sections describe what you should enter in each area of the parameter window.

COMI cell name

In order to make the cell easy to identify, include the name of the action template the cell calls out. The cell name shown in Figure 4-3 is “start getbalance transaction.”

Call Audit Enabled

Determines if this cell logs the following information to the call audit statistics file (audit_stat.d):

- Application Name
- Cell Name
- Cell Number
- Date and Time of Cell Execution
- Contents of the Cell Comment field
- Contents of the Call Audit Information buffer

The default setting is No.

Call Audit Information

When you enable Call Auditing, the Call Auditing process logs the contents of this buffer to the audit_stat.d file. The default setting is DIGITS.

COMI action template

Enter the name of the action template, in quotes, and leave off the .act suffix. Figure 4-3 shows “getbalance” as the entered name.

COMI cell more input

If you need to send more than 10 input buffers to the host computer, click the button in front of “Yes”; if you are sending 10 or less input buffers, click the button in front of “No.” If you specify “Yes,” place additional COMI cells on the drawing board until you have enough input buffers. Connect the Success branch of the first COMI cell to the Input branch of the following COMI cell. Connect the rest of the COMI cells in this fashion. The final COMI cell should have the “No” button selected for the “More Input?” parameter.

COMI cell timeout

Select the number of seconds you want the TRS process to wait for the transaction to complete by moving the slider under “Timeout”. There is an available range from 1 to 75 seconds. This indicates the maximum time for the COMI cell to start the transaction and send the necessary input buffers, and for the COMO cell to receive the output buffers. The TRS process rejects the transaction and sends back a timeout if the action is not performed within the time allotted.

Note: The application will take the timeout branch of the COMI cell if a timeout occurs.

COMI cell buffer count

You do not need to enter a value in this field as Meridian IVR 2.0/I automatically calculates the number of buffers when you click the “Apply” button.

COMI cell input buffers

Enter the names of the input buffers containing the information to pass to the TRS process. The input buffer shown in Figure 4-3 is ACCOUNTNUMBER, without any quotes. In this example, you must program your application to place the customer’s account number in the ACCOUNTNUMBER buffer before the COMI cell is executed.

In this example, ACCOUNTNUMBER is a user-defined buffer; you can use any name for your applications, or you can use system buffers.

Note: You must enter the input buffers in the same order the host computer uses them.

If you need to use more than 10 input buffers in your application, make sure you select “Yes” for “More Input?”, then string COMI cells together until you have enough input buffers. Connect each additional COMI cell to the previous one’s “SUCCESS” branch.

Setting the COMO cell parameters

Once you have set the COMI cell to send information, code the COMO cell to receive information. You must place a COMO cell directly after the COMI cell to complete the transaction, even if the host computer is not sending any data to any output buffers. The TRS process sends verification that the transaction completed successfully or notification that there was an error in the COMO cell. [Figure 4-4](#) shows the parameter window for the COMO cell.

Figure 4-4
COMO cell parameter window

COMO Parameters

Call ID: **COMO** Output from Host

Receive Balance

Comments

Call Audit Enabled? ☐ Yes ☒ No

Call Audit Information: GITS Yes

Blocking? ☒ Yes ☐ No

Output Buffers

Buffer Count: 0 Yes

1. Yes

2. Yes

3. Yes

4. Yes

5. Yes

6. Yes

7. Yes

8. Yes

9. Yes

10. Yes

Apply Cancel Help

The following sections describe what you should enter in each area of the parameter window.

COMO cell name

The cell name shown in Figure 4-4 is “Receive Balance.” The name helps identify the function of the cell.

Call Audit Enabled

Determines if this cell logs the following information to the call audit statistics file (audit_stat.d):

- Application Name
- Cell Name
- Cell Number
- Date and Time of Cell Execution
- Contents of the Cell Comment field
- Contents of the Call Audit Information buffer

The default setting is No.

Call Audit Information

When you enable Call Auditing, the Call Auditing process logs the contents of this buffer to the audit_stat.d file. The default setting is DIGITS.

COMO cell blocking?

Activating blocking tells the COMO cell to wait for the transaction to complete before continuing to the next cell. The application will wait until the TRS interaction ends before it continues through the Meridian IVR 2.0/I application (only one of the branches “MORE DATA” or “END OF DATA” is to be connected to the subsequent cell). If you select “No,” the COMO cell receives a status code from the TRS process. If the transaction is complete, the application follows the “END OF DATA” or “MORE DATA” branch (whichever one connects to another cell). If the transaction is not complete, the application follows the “NOT READY” branch. You could use this branch to play a prompt notifying the caller that the transaction is in progress, and then feed the call flow back into the COMO cell.

COMO cell buffer count

You do not need to enter a value in this field as Meridian IVR 2.0/I automatically calculates the number of buffers when you click on the “Apply” button.

COMO cell output buffers

Enter the name of the buffers that will accept the output from the host computer. Figure 4-4 shows BALANCE (no quotes) as the only output buffer for this example.

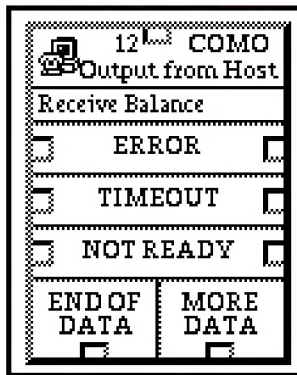
Note: You must enter the output buffers in the same order they will be used by the TRS process.

If your application uses more than 10 output buffers, string COMO cells together until you have enough buffers. Connect the input branch of each subsequent COMO cells to the “MORE DATA” branch of the previous COMO cell.

COMO cell branches

As shown in [Figure 4-5](#), the COMO cell has several branches.

Figure 4-5
COMO cell



12 COMO	
Output from Host	
Receive Balance	
☐	ERROR
☐	TIMEOUT
☐	NOT READY
☐	END OF DATA
☐	MORE DATA

Table 4-1 describes each branch of the COMO cell.

Table 4-1
Branches of COMO cell

Branch	Reason Taken
ERROR	Any problem with any part the host transaction will cause the application to take the error branch.
TIMEOUT	The transaction took more than its allowed time as specified in the COMI cell.
NOT READY	The transaction did not complete and Blocking was set to "No".
END OF DATA	The transaction successfully completed.
MORE DATA	The transaction requires additional output buffers.

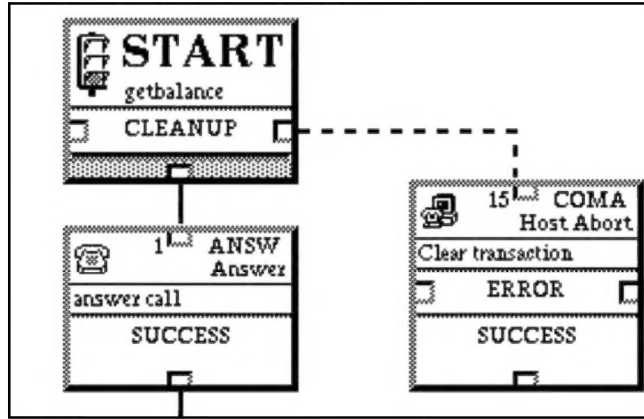
For each COMO cell, use either the "END OF DATA" branch, or the "MORE DATA" branch, but not both. If you use the "MORE DATA" branch, the next cell must be another COMO cell. The last COMO cell should use the "END OF DATA" branch.

Setting the COMA cell parameters

Use a COMA cell in the clean-up branch of the START cell to deal with an error like a caller hanging up in the middle of a transaction. See [Figure 4-6](#).

If you selected manual mode in the application template, use the COMA cell to abort a transaction in progress and release the session (see Chapter 2).

Figure 4-6
COMA cell in the clean-up handler



The COMA cell frees all memory and buffers associated with the COMI and COMO cells.

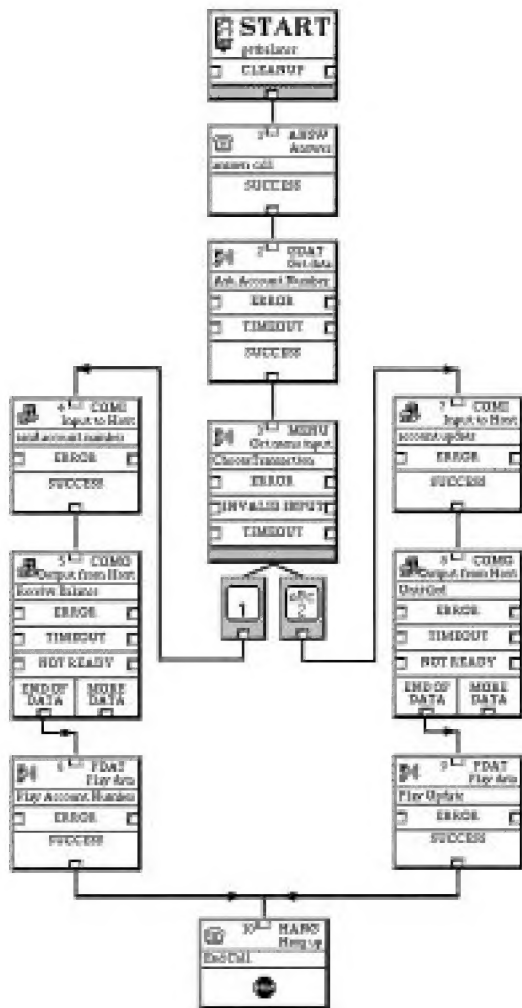
The COMA cell parameters window contains a cell name field, and a comments field. The COMA cell in [Figure 4-6](#) has “Clear transaction” as its cell name.

Sample application using the COMI, COMO, and COMA cells

[Figure 4-7](#) shows a sample Meridian IVR 2.0/I application that uses the COMI, COMO, and COMA cells to retrieve a customer’s account balance stored on a host computer. This example application uses the action and screen templates created in Chapter 3. A caller activates this example application by pressing “1” after hearing the prompt played by the MENU cell. A separate action template and its associated screen templates would need to be created to support menu choice 2.

To illustrate the logic of the call flow, these applications are shown as one page applications with no error branches. In reality, these applications would span several pages and would have all error branches connected appropriately.

Figure 4-7
Application accessing the TRS process from the COM1 and COMO cell



Prior to the execution of this application, the initial-action template is executed (when Meridian IVR 2.0/I is started on the application processor). Once a call from a customer is received, the Meridian IVR 2.0/I application performs these steps:

Table 4-2
Application cell functions for Figure 4-7

Cell 1	The ANSW cell answers the incoming call.
Cell 2	The GDAT cell plays a prompt requesting the caller to enter an account number. The number entered is stored in the ACCOUNTNUMBER buffer.
Cell 3	The MENU cell plays a prompt instructing the caller to press 1 to get an account balance, or press 2 to get an account update.
Cell 4	When the caller presses "1" on the telephone, the COMI cell starts the transaction. The action template specified in the COMI cell parameter window is "getbalance," - the same template used in the example transaction described in Chapter 3. The COMI cell provides the contents of the ACCOUNTNUMBER buffer to the host computer application.
Cell 5	The COMO cell receives the customer's balance from the BALANCE buffer.
Cell 6	The contents of the BALANCE buffer is played to the user.
Cell 7-9	These cells support menu choice 2, which provides an account update in a similar manner.
Cell 10	The caller hangs up and the application ends.

If the caller hangs up before the transaction is completed, the COMA cell in the clean-up branch of the START cell clears the memory and buffers associated with the application.

Figure 4-8 shows how the COMI and COMO cell parameters relate to the screens defined on the host computer. This transaction uses an action template, getbalance, to call three screen templates - one to choose the Accounts Receivable menu, one to enter the account number, and one for the customer information screen. Each screen template lists the fields that will be accessed.

Figure 4-8
COMI, COMO cell parameters, TRS templates, and host screens

